



# Tapping the Power in PowerShell

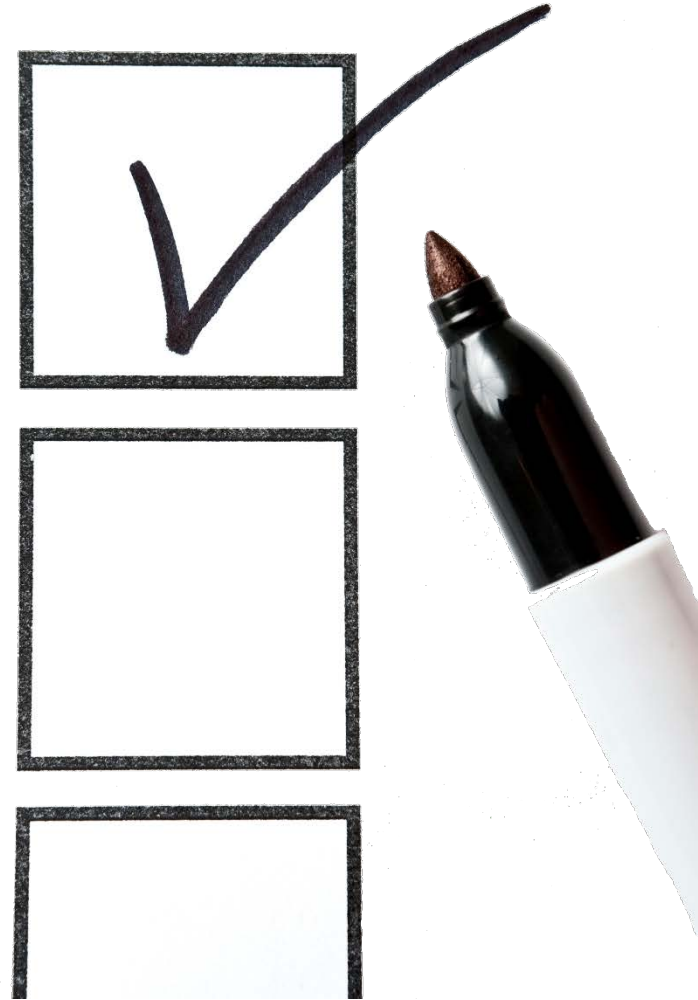
Sean P. McDonough  
Chief Technology Officer and Owner, Bitstream Foundry LLC  
Microsoft MVP (Office Servers and Services)



[www.bitstreamfoundry.com](http://www.bitstreamfoundry.com)

# Agenda

- Initial Words
- Scripting for SharePoint Online
- DSC and SharePoint Configuration
- A Quick Look at Newer Tricks and Techniques (PowerShell 4 & 5)
- Conclusion and Wrap-Up

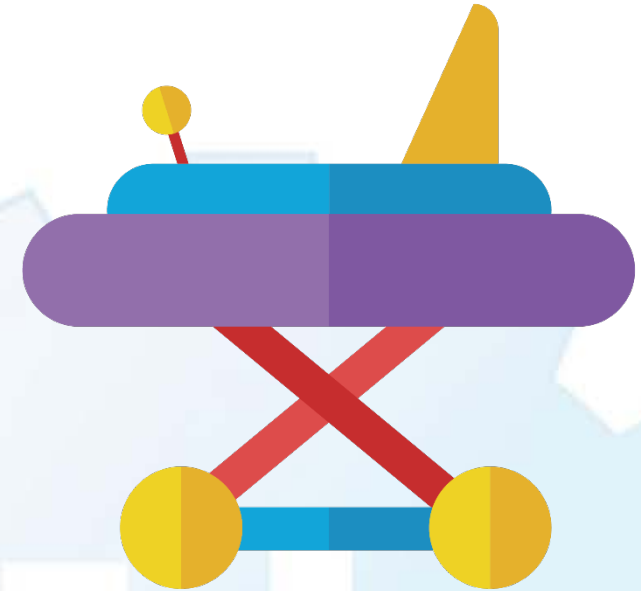


# Initial Words



# About This Talk

- No (minimal) training wheels
  - Good link in references for learning
  - Remember to use **Get-He1p** for syntax
- Focus on new(er) PowerShell capabilities
  - Lots of Goodness in PowerShell 4 & 5
  - What's useful to SharePoint administrators *and* developers
- A variety of script samples and walk-throughs
  - Using Powershell Integrated Scripting Environment (ISE)
  - For both on-premises and SharePoint Online



# Setting Up Your Environment

- You'll need a few things
  - A SharePoint environment (could be SPO)
  - An editor; e.g., PowerShell ISE or VS Code
  - Windows Management Framework 5.0
  - A willingness to explore and **see red**
- Connecting to SPO?
  - Presents some slightly different challenges versus on-premises scripting
  - Some cmdlet differences
  - We'll shift to cover this in a bit

Operating System	Editions	Prerequisites	Package Links
Windows Server 2012 R2			<a href="#">Win8.1AndW2K12R2-KB3134758-x64.msu</a>
Windows Server 2012			<a href="#">W2K12-KB3134759-x64.msu</a>
Windows Server 2008 R2 SP1	All, except IA64	WMF 4.0 and .NET Framework 4.5 or above are installed	<a href="#">Win7AndW2K8R2-KB3134760-x64.msu</a>
Windows 8.1	Pro, Enterprise		<b>x64:</b> <a href="#">Win8.1AndW2K12R2-KB3134758-x64.msu</a> <b>x86:</b> <a href="#">Win8.1-KB3134758-x86.msu</a>
Windows 7 SP1	All	WMF 4.0 and .NET Framework 4.5 or above are installed	<b>x64:</b> <a href="#">Win7AndW2K8R2-KB3134760-x64.msu</a> <b>x86:</b> <a href="#">Win7-KB3134760-x86.msu</a>

# Good Hygiene and Technique

- Avoid an “Oops”
  - The question isn’t whether or not you have a test environment – it’s whether or not you have a production environment
  - Use **-whatif** on test runs if available with cmdlets
  - Where possible, build test/trial-run capabilities into your scripts if they’re going to be making changes
- Be a Good Admin and Digital Citizen
  - Add a header block and commenting to scripts
  - Enhances the **get-help** experience for users

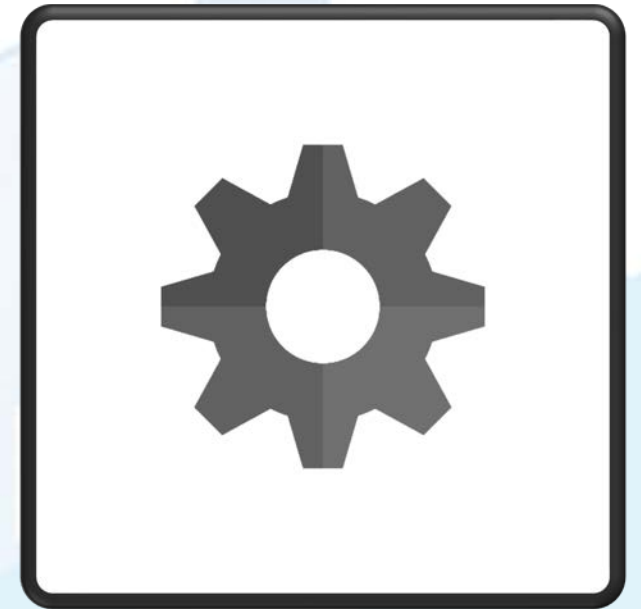
```
1 <#
2
3 .SYNOPSIS
4 Assigns licenses to one or more users in an Office 365 tenant
5 based on selective criteria.
6
7 .DESCRIPTION
8 The default action for this script is to assign a selected
9 license type to any user in a tenant who (a) isn't disabled, and
10 (b) doesn't already have a license assigned. Several switches
11 allow the user to manually specify an inclusion or exclusion
12 list to control users who are affected by the script's actions.
13 In addition, a parameter exists to perform the equivalent of a
14 "whatif" with the script to identify how each account in the
15 processing list would be treated. Whether executed in test mode
16 or not, the script will write out its actions (or expected
17 actions) into a CSV file.
18
19 .PARAMETER tenantName
20 The name of the Office 365 tenant (e.g., contoso) against which
21 actions should be taken.
22
23 .PARAMETER licenseType
24 The name (as recognized by Office 365) of the license to be
25 assigned. For a reference of individual skus, look online (e.g.,
26 http://adaptivedge.com/2016/04/13/granular-licensing-of-office-36
27 For more common skus, consider ENTERPRISEPACK for Office 365 E3
28 and ENTERPRISEPREMIUM for Office 365 E5, and
29 O365_BUSINESSPREMIUM for Small Business Premium.
30
31 .PARAMETER inclusionCSV
32 A path to a CSV that contains the specific accounts to which
33 licenses should be assigned. The CSV must contain a column with
34 a header of "UPN" (no quotes) that specifies the accounts for
35 inclusion by user principal name (e.g., bob@contoso.com). This
36 parameter cannot be used with the -exclusionCSV parameter.
37
38 .PARAMETER exclusionCSV
39 A path to a CSV that contains the specific accounts to which
40 licenses should not be assigned; i.e., licenses will be assigned
41 to all accounts except those that appear in this file. The CSV
```

# Scripting for SharePoint Online



# Connecting to SPO

- Preparing to talk to SPO
  - You'll need a few dependencies
    - Microsoft Online Services Sign-in Assistant
    - SharePoint Online Management Shell
    - Windows Azure AD Directory Module (recommended)
  - Others may also include
    - Exchange Online cmdlets
    - Security & Compliance Center cmdlets
    - Skype for Business Online cmdlets
    - See references at end for links





# Making the Connection

- Connect to SPO

- Easy to do by itself (replace **<yourTenantName>** below)

```
1 Import-Module Microsoft.Online.Sharepoint.PowerShell
2 $credential = Get-credential
3 Connect-SPOService -url https://<yourTenantName>-admin.sharepoint.com -Credential $credential
```

- Consider connecting to all related services

- Usually need more than just an SPO connection
- Azure Active Directory and SPO go hand-in-hand
- Try to remember to close out remote sessions when done
- See **ConnectToMso1Services.ps1** script for connect/disconnect assistance

# Common SPO Activities

- License Reporting
  - Strictly speaking, not an SPO activity (it's Azure AD)
  - Repetitive tasks, so PowerShell is a natural fit
  - DEMO: **GetMso1UsersAndState.ps1**
- (Conditional) License Assignment
  - Assign a license to everyone who doesn't have one
  - Conditionally assign licenses to a subset of users – either specific users (inclusion group) or all users except a specific group (exclusion group)
  - DEMO: **AssignMso1UserLicense.ps1**

```
19 .LINK
20 http://SharePointInterface.com
21
22 #>
23
24 # Prompt for credentials
25 $credential = Get-Credential
26
27 # Setup credentials and import necessary module
28 Write-Output ""
29 Write-Output "Opening and connecting to MSONline ..."
30 Import-Module MsOnline
31 (Connect-MsolService -Credential $credential) > $null
32
33 File Name : AssignMso1UserLicense.ps1
34 Author : Sean McDonough - sean@sharepointi
35 Last Update: December 3, 2016
36
37 .LINK
38 http://SharePointInterface.com
39
40 #>
41 param(
42 [Parameter(Mandatory=$true)]
43 [String]$tenantName,
44 [Parameter(Mandatory=$true)]
45 [String]$licenseType,
46 [String]$inclusionCSV="",
47 [String]$exclusionCSV="",
48 [Boolean]$isTest=$true
49 )
50 cls
51
52 # Setup some variables we'll be using. We need to
53 # Just in case the script is run multiple times (w
54 $usingExclusions = $false
55 $usingInclusions = $false
56 $userList = {}
```

# Common SPO Activities

- (Sub-)site Creation
  - PowerShell represents a much quicker and efficient way to bulk-create sites and sub-sites
  - Highlight a few different techniques and concepts that are important for scripts:
    - A more developer-oriented approach to PowerShell scripting
    - Leveraging SharePoint's Client-Side Object Model (CSOM)
    - Structured exception handling
  - DEMO: **CreateSpoSubSites.ps1**

```
142 # Create the subsite if not in test mode
143 $reason = ""
144 if ($isTest -eq $true)
145 {
146     Write-Output "- Running in test mode
147     $reason = "Test mode is TRUE; skippi
148 } else {
149     # Setup the necessary site host cont
150     Write-Output "- Specifying subsite h
151     $context = New-Object Microsoft.Shar
152     $context.Credentials = $spoCredentia
153
154     # Prep and execute CSOM to create su
155     Write-Output "- Performing subsite c
156     $targetSubsite = $context.Web.Webs.A
157     $context.ExecuteQuery()
158
159     # Sub-site creation successful
160     Write-Output "- Subsite creation suc
161     $reason = "Test mode is FALSE; subsi
162     $subsitesCreatedCount += 1
163 }
164
165 } catch {
166
167     #Something went wrong. Indicate a proble
168     Write-Output "- SUBSITE CREATION FAILED!
169     $reason = $_.Exception.Message
```

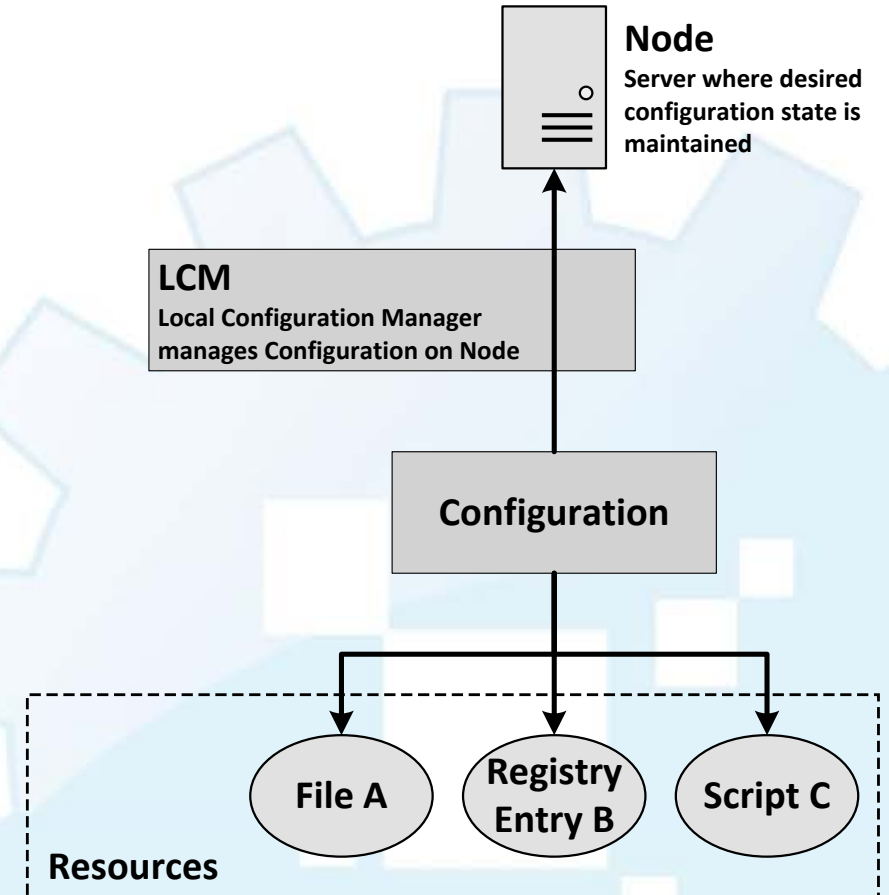
# DSC and SharePoint Configuration



# What is DSC?

## DSC = Desired State Configuration

- A new way of managing the configuration on nodes (computers, VMs, services, etc.) using the following types of components:
  - **Configurations.** Declarative PowerShell scripts which define and configure instances of resources.
  - **Resources.** Building blocks which model various components of a system (e.g., file, service, Registry, etc.)
  - **Local Configuration Manager.** The engine that facilitates interactions between configurations and resources to impact VMs, services, etc.



# What Can DSC Do?

- Ensures that the nodes in your data center have the correct configuration
  - Allows for continuous deployment
  - Prevents configuration drift
  - Very important: *operates in an idempotent manner!*
- Some examples of what DSC can do
  - Manage groups and user accounts
  - Deploy software
  - Enable and disable server roles and features
  - Manage files and directories
  - Manage Registry settings

```
1 Configuration DscConfigDemo1
2 {
3     Import-DscResource -ModuleName 'PSDesiredStateConfiguration'
4     Node "SP2013-DEV"
5     {
6
7         File DeploymentFolder
8         {
9             Ensure = "Present"
10            Recurse = $true
11            Type = "Directory"
12            SourcePath = "E:\Tapping the Power in the Cloud"
13            DestinationPath = "E:\DscDemoRun1"
14        }
15
16        File ConfigXml
17        {
18            Ensure = "Present"
19            Type = "File"
20            SourcePath = "E:\Tapping the Power in the Cloud"
21            DestinationPath = "E:\DscDemoRun1\Test\config.xml"
22            DependsOn = "[File]DeploymentFolder"
23        }
24
25        File SampleFile
26        {
27            Ensure = "Present"
28        }
29    }
30 }
```

# Using DSC 101

1. Create a configuration script (e.g., DscConfigDemo1.ps1)
2. Execute configuration script to generate MOF file
3. Review/Update LocalConfigurationManager settings
  - `Get-DscLocalConfigurationManager`
  - Build LCM configuration script and generate special MOF
  - `Set-DscLocalConfigurationManager`
4. Apply the configuration script/MOF
  - `Start-DscConfiguration`
5. Monitor the configuration actions

**DEMO: DscConfigDemo1.ps1**

# DSC and SharePoint

## What about SharePoint?

- DSC is highly extensible beyond the basic resources provided
- The SharePointDsc project extends DSC to include a wealth of new SharePoint resources for use in configuration management
- DEMOS: `DscConfigDemo2.ps1`, `DscConfigDemo3.ps1`

```
PS E:\Tapping the Power in PowerShell\dsc> Get-DscResource -Module SharePointDsc
```

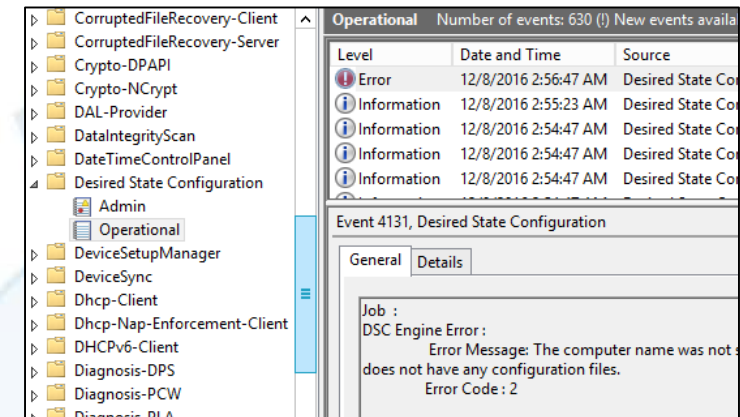
ImplementedAs	Name	ModuleName	Version	Properties
PowerShell	SPAccessServiceApp	SharePointDSC	1.4.0.0	{ApplicationPool, DatabaseServer, Name, Depends...
PowerShell	SPAlternateUrl	SharePointDSC	1.4.0.0	{WebAppUrl, Zone, DependsOn, Ensure...}
PowerShell	SPAntivirusSettings	SharePointDSC	1.4.0.0	{ScanOnDownload, AllowDownloadInfected, Attempt...
PowerShell	SPAppCatalog	SharePointDSC	1.4.0.0	{SiteUrl, DependsOn, InstallAccount, PsDscRunAs...
PowerShell	SPAppDomain	SharePointDSC	1.4.0.0	{AppDomain, Prefix, DependsOn, InstallAccount...}
PowerShell	SPAppManagementServiceApp	SharePointDSC	1.4.0.0	{ApplicationPool, Name, DatabaseName, DatabaseS...
PowerShell	SPAppStoreSettings	SharePointDSC	1.4.0.0	{WebAppUrl, AllowAppPurchases, AllowAppsForOffi...
PowerShell	SPBCSServiceApp	SharePointDSC	1.4.0.0	{ApplicationPool, Name, DatabaseName, DatabaseS...
PowerShell	SPBlobCacheSettings	SharePointDSC	1.4.0.0	{EnableCache, WebAppUrl, Zone, DependsOn...}
PowerShell	SPCacheAccounts	SharePointDSC	1.4.0.0	{SuperReaderAlias, SuperUserAlias, WebAppUrl, D...
PowerShell	SPConfigWizard	SharePointDSC	1.4.0.0	{Ensure, DatabaseUpgradeDays, DatabaseUpgradeTi...
PowerShell	SPContentDatabase	SharePointDSC	1.4.0.0	{Name, WebAppUrl, DatabaseServer, DependsOn...}
PowerShell	SPCreateFarm	SharePointDSC	1.4.0.0	{AdminContentDatabaseName, DatabaseServer, Farm...
PowerShell	SPDatabaseAAG	SharePointDSC	1.4.0.0	{AGName, DatabaseName, DependsOn, Ensure...}
PowerShell	SPDesignerSettings	SharePointDSC	1.4.0.0	{SettingsScope, Url, AllowCreateDeclarativeWork...
PowerShell	SPDiagnosticLoggingSet...	SharePointDSC	1.4.0.0	{LogPath, LogSpaceInGB, AppAnalyticsAutomaticUp...
PowerShell	SPDistributedCacheService	SharePointDSC	1.4.0.0	{CacheSizeInMB, CreateFirewallRules, Name, Serv...
PowerShell	SPExcelServiceApp	SharePointDSC	1.4.0.0	{ApplicationPool, Name, CachingOfUnusedFilesEna...
PowerShell	SPFarmAdministrators	SharePointDSC	1.4.0.0	{Name, DependsOn, InstallAccount, Members...}
PowerShell	SPFarmSolution	SharePointDSC	1.4.0.0	{LiteralPath, Name, DependsOn, Deployed...}
PowerShell	SPFeature	SharePointDSC	1.4.0.0	{FeatureScope, Name, Url, DependsOn...}
PowerShell	SPHealthAnalyzerRuleState	SharePointDSC	1.4.0.0	{Enabled, Name, DependsOn, FixAutomatically...}
PowerShell	SPInstall	SharePointDSC	1.4.0.0	{BinaryDir, ProductKey, DataPath, DependsOn...}
PowerShell	SPInstallLanguagePack	SharePointDSC	1.4.0.0	{BinaryDir, BinaryInstallDays, BinaryInstallTim...
PowerShell	SPInstallPrereqs	SharePointDSC	1.4.0.0	{InstallerPath, OnlineMode, AppFabric, DependsO...
PowerShell	SPIrmSettings	SharePointDSC	1.4.0.0	{Ensure, DependsOn, InstallAccount, PsDscRunAsC...
PowerShell	SPJoinFarm	SharePointDSC	1.4.0.0	{DatabaseServer, FarmConfigDatabaseName, Passph...
PowerShell	SPManagedAccount	SharePointDSC	1.4.0.0	{AccountName, Account, DependsOn, EmailNotifica...
PowerShell	SPManagedMetaDataServi...	SharePointDSC	1.4.0.0	{ApplicationPool, Name, ContentTypeHubUrl, Data...
PowerShell	SPManagedPath	SharePointDSC	1.4.0.0	{Explicit, HostHeader, RelativeUrl, WebAppUrl...}
PowerShell	SPOfficeOnlineServerBi...	SharePointDSC	1.4.0.0	{DnsName, Zone, DependsOn, Ensure...}
PowerShell	SPOutgoingEmailSettings	SharePointDSC	1.4.0.0	{CharacterSet, FromAddress, ReplyToAddress, SMT...
PowerShell	SPPasswordChangeSettings	SharePointDSC	1.4.0.0	{MailAddress, DaysBeforeExpiry, DependsOn, Inst...
PowerShell	SPPerformancePointServ...	SharePointDSC	1.4.0.0	{ApplicationPool, Name, DatabaseName, DatabaseS...
PowerShell	SPProductUpdate	SharePointDSC	1.4.0.0	{SetupFile, BinaryInstallDays, BinaryInstallTim...



# The (Slightly) Darker Side of DSC

## Some troubleshooting pointers ...

- Want to understand what's happening under-the-hood?
  - *Event Viewer > Applications and Services Logs > Microsoft > Windows > Desired State Configuration*
  - Excellent way to determine why you're seeing "Failure" messages when you apply a configuration
- DSC stores its files in **C:\Windows\System32\Configuration**
  - If you "break" your system playing around, you can wipe the MOF files in this folder and all the MOF + JSON files in the **ConfigurationStatus** subfolder.
- Like everything else (with SharePoint) – chances are the problem is permissions!



The screenshot shows the contents of the C:\Windows\System32\Configuration folder. The table below lists the files and folders:

Name	Date modified	Type	Size
BaseRegistration	12/3/2016 5:00 PM	File folder	
ConfigurationStatus	12/8/2016 3:04 AM	File folder	
PartialConfigurations	12/3/2016 5:00 PM	File folder	
Registration	8/22/2013 11:39 AM	File folder	
Schema	8/22/2013 11:39 AM	File folder	
DSCEngineCache.mof	12/8/2016 3:04 AM	MOF File	1 KB
DSCResourceStateCache.mof	12/8/2016 3:04 AM	MOF File	1 KB
DSCStatusHistory.mof	12/8/2016 3:04 AM	MOF File	2 KB
MetaConfig.backup.mof	12/8/2016 3:04 AM	MOF File	2 KB
MetaConfig.mof	12/8/2016 3:04 AM	MOF File	2 KB

# A Quick Look at Newer Tricks and Techniques (PowerShell 4 & 5)



# So ... What's New?

Entirely too much to cover in the time we have left!

- I'm going to pick a few of the bigger items to call out
- For an exhaustive list, see the references at the end of this presentation

Some of the additions that caught my eye:

- DSC (which we already discussed)
- **Class**, **Enum**, and other object-oriented keywords
- Windows PowerShell Workflow
- **ConvertFrom-String** cmdlet
- PowerShell Gallery integration with **Find-Module**, **Find-Script**, etc.

Don't be afraid to explore and tinker!

# Wrap-Up

## PowerShell isn't just a simple scripting language

- Recent PowerShell 4 & 5 additions have expanded PowerShell's range of usefulness to workflow, configuration management, online management, and more.

## PowerShell isn't just for IT Pros

- Many of the newer features may even be more attractive to developers
- If you're an IT Pro, expand your reach and capabilities. A little dev never hurts!

## Above all, don't be afraid to dig-in!

- Play with the scripts I'm sharing with you
- Find a problem or two to solve with PowerShell in your day-to-day work

# Contact



# Sean P. McDonough

SharePoint and Office 365 Gearhead,  
Tinkerer, Microsoft MVP

My Company:



Twitter:

@spmcdonough

Blog:

<http://SharePointInterface.com>

About:

<http://about.me/spmcdonough>



# References

## Weekend Scripter: The Best Ways to Learn PowerShell

- <https://blogs.technet.microsoft.com/heyscriptingguy/2015/01/04/weekend-scripter-the-best-ways-to-learn-powershell/>

## Visual Studio Code

- <https://code.visualstudio.com/>

## Announcing PowerShell language support for Visual Studio Code and more!

- <https://blogs.msdn.microsoft.com/powershell/2015/11/16/announcing-powershell-language-support-for-visual-studio-code-and-more/>

## Make a Simple Change to PowerShell to Prevent Accidents

- <https://blogs.technet.microsoft.com/heyscriptingguy/2011/11/21/make-a-simple-change-to-powershell-to-prevent-accidents/>

## Windows Management Framework 5.0 Download

- <https://www.microsoft.com/en-us/download/details.aspx?id=50395>

# References

## WMF 5.0 Installation Details

- <https://msdn.microsoft.com/en-us/powershell/wmf/5.0/requirements>

## Add help to your own PowerShell scripts

- <http://kevinpelgrims.com/blog/2010/05/10/add-help-to-your-own-powershell-scripts/>

## Office 365 PowerShell for SharePoint Online

- <https://technet.microsoft.com/en-us/library/fp161362.aspx>

## Connect to Office 365 PowerShell

- <https://technet.microsoft.com/en-us/library/dn975125.aspx>

## SharePoint Online Management Shell

- <https://www.microsoft.com/en-us/download/details.aspx?id=35588>

# References

Connect to all Office 365 services in a single Windows PowerShell window

- <https://technet.microsoft.com/en-us/library/dn568015.aspx>

Cmdlet references for Office 365 services

- <https://technet.microsoft.com/en-us/library/dn789037.aspx>

Office 365 – Creating a Subsite (Web) using CSOM in SharePoint Online

- <https://blogs.technet.microsoft.com/fromthefield/2015/01/26/office-365-creating-a-subsite-web-using-csom-in-sharepoint-online/>

About Try Catch Finally

- [https://msdn.microsoft.com/en-us/powershell/reference/5.0/microsoft.powershell.core/about/about\\_try\\_catch\\_finally](https://msdn.microsoft.com/en-us/powershell/reference/5.0/microsoft.powershell.core/about/about_try_catch_finally)

RESTful web-services to get and search Countries

- <http://www.groupkt.com/post/c9b0ccb9/country-and-other-related-rest-webservices.htm>



# References

## Using the SharePoint REST Service with Windows PowerShell

- <https://www.itunity.com/article/sharepoint-rest-service-windows-powershell-1381>

## Get to know the SharePoint 2013 REST service

- <https://msdn.microsoft.com/en-us/library/office/fp142380.aspx>

## Online JSON Viewer

- <http://jsonviewer.stack.hu/>

## Windows PowerShell Desired State Configuration Overview

- <https://msdn.microsoft.com/en-us/powershell/dsc/overview>

## Desired State Configuration in Windows Server 2012 R2 PowerShell

- <https://channel9.msdn.com/Events/TechEd/Europe/2013/MDC-B302#fbid=mhhzaiP6RCb>

# References

xSharePoint is now SharePointDsc – what you need to know!

- [https://blogs.msdn.microsoft.com/brian\\_farnhill/2016/06/08/xsharepoint-is-now-sharepointdsc-what-you-need-to-know/](https://blogs.msdn.microsoft.com/brian_farnhill/2016/06/08/xsharepoint-is-now-sharepointdsc-what-you-need-to-know/)

SharePointDsc on GitHub

- <https://github.com/PowerShell/SharePointDsc>

Using Credentials with PsDscAllowPlainTextPassword and PsDscAllowDomainUser in PowerShell DSC Configuration Data

- <https://blogs.technet.microsoft.com/ashleymcglone/2015/12/18/using-credentials-with-psdscallowplaintextpassword-and-psdscallowdomainuser-in-powershell-dsc-configuration-data/>

Troubleshooting DSC

- <https://msdn.microsoft.com/en-us/powershell/dsc/troubleshooting>

# References

## What's New with PowerShell

- <https://msdn.microsoft.com/en-us/powershell/scripting/whats-new/what-s-new-with-powershell>

## Introduction to PowerShell 5 Classes

- <https://blogs.technet.microsoft.com/heyscriptingguy/2015/08/31/introduction-to-powershell-5-classes/>

## PowerShell 5: Create Simple Class

- <https://blogs.technet.microsoft.com/heyscriptingguy/2015/09/01/powershell-5-create-simple-class/>

## Getting Started with Windows PowerShell Workflow

- [https://technet.microsoft.com/en-us/library/jj134242\(v=ws.11\).aspx](https://technet.microsoft.com/en-us/library/jj134242(v=ws.11).aspx)

## Learning Windows PowerShell Workflow

- <https://docs.microsoft.com/en-us/azure/automation/automation-powershell-workflow>

# References

## Using the ConvertFrom-String cmdlet to parse structured text

- <http://www.powershellmagazine.com/2014/09/09/using-the-convertfrom-string-cmdlet-to-parse-structured-text/>

## PowerShell Gallery

- <https://www.powershellgallery.com/>

## Find-Module

- [https://msdn.microsoft.com/en-us/powershell/gallery/psget/module/psget\\_find-module](https://msdn.microsoft.com/en-us/powershell/gallery/psget/module/psget_find-module)

# Feedback



Please provide feedback!

Let us know what you think!

# Contact



# Sean P. McDonough

SharePoint and Office 365 Gearhead,  
Tinkerer, Microsoft MVP

My Company:



Twitter:

@spmcdonough

Blog:

<http://SharePointInterface.com>

About:

<http://about.me/spmcdonough>

